

Non-iterative Type 4 and 5 Nonuniform FFT Methods in the One-Dimensional Case

J. Selva

Abstract—The so-called non-uniform FFT (NFFT) is a family of algorithms for efficiently computing the Fourier transform of finite-length signals, whenever the time or frequency grid is nonuniformly spaced. Following the usual classification, there exist five NFFT types. Types 1 and 2 make it possible to pass from the time to the frequency domain with nonuniform input and output grids respectively. Type 3 allows for both input and output nonuniform grids. Finally, types 4 and 5 are the inverses of types 1 and 2 and are expensive computationally, given that they involve iterative methods. In this paper, we solve this last drawback in the one-dimensional case by presenting non-iterative type 4 and 5 NFFT methods that just involve three NFFTs of types 1 or 2 plus some additional FFTs. The methods are based on exploiting the structure of the Lagrange interpolation formula. The paper includes several numerical examples in which the proposed methods are compared with the Gaussian elimination (GE) and conjugate gradient (CG) methods, both in terms of round-off error and computational burden.

I. INTRODUCTION

As is well known, the FFT computes the spectrum of a finite-length discrete signal with complexity $O(P \log P)$, where P is the signal's length. However, the FFT requires regular grids in both the time (or spatial) and frequency domains. This constraint is a shortcoming in various applications in which the sampling or evaluation grid is nonuniform. The so-called nonuniform FFT (NFFT) is a family of algorithms for performing the same transform as the FFT but with irregular input or output grids. The NFFT has been developed during the last decades for various applications [1]–[3]. There exist several surveys and tutorials on this topic [4]–[7].

Following the classification in [8], there are five basic types of NFFT. The first three types have the same complexity order as the FFT and are non-iterative, while types 4 and 5 require an iterative procedure like the conjugate gradient (CG) method, [9, Sec. 11.3]. Type 1 performs the same operation as the FFT but with input samples taken nonuniformly, while type 2 is similar to the inverse FFT but with nonuniform output instants (or positions). Type 3 is a combination of types 1 and 2 and can be viewed as a method to compute the spectrum of a nonuniform delta train at nonuniform frequencies. Finally, types 4 and 5 are the inverses of types 1 and 2 respectively, and are computed iteratively, usually through the CG method, thus being significantly more expensive computationally than the first three types. Concretely, each iteration of the CG method for either type 4 or 5 has the same complexity order as the FFT, but the number of iterations is large.

The purpose of this paper is to present two non-iterative methods for types 4 and 5 in the one-dimensional case, which exploit the properties of the Lagrange interpolation formula. They have the same complexity order as the FFT, $O(P \log P)$,

and are far less complex than the existing CG implementations. The extension of the proposed methods to multiple dimensions seems difficult, given that there is no general Lagrange formula in several variables [10].

The paper has been organized as follows. In the next section, we shortly recall the five NFFT types, and introduce the Lagrange interpolation formula. Then, we present the proposed NFFT methods in Secs. III and IV. These methods depend on two parameters, the attenuation and oversampling factors, which are analyzed in Sec. V. Finally, Sec. VI contains a numerical assessment of the methods' round-off error and complexity.

A. Notation

In the rest of the paper, we will use the following notation:

- Definitions will be introduced using the symbol “ \equiv ”.
- “ $O(\cdot)$ ” will be the big-O notation.
- P , Q , and R will denote positive integers.
- The notation $\{\cdot\}_p^P$ will represent the vector formed by evaluating the expression inside curly braces for $p = 0, \dots, P-1$. Thus, for a function $f(x)$, $\{f(p)\}_p^P$ will be the vector $[f(0), f(1), \dots, f(P-1)]$.
- The operators “DFT” and “IDFT” will respectively denote the DFT and IDFT of a vector. Thus, given a sequence v_q , $\text{DFT}\{v_q\}_q^P$ is the P -length vector whose element at position $p+1$ is

$$\sum_{q=0}^{P-1} v_q e^{-j2\pi pq/P}. \quad (1)$$

- “ \odot ” will represent the element-by-element product of two vectors,

$$\{v_p\}_p^P \odot \{w_p\}_p^P = \{v_p w_p\}_p^P. \quad (2)$$

- The operator “Coef” will extract the coefficient vector of a given trigonometric polynomial,

$$\text{Coef} \sum_{p=0}^{P-1} F_p e^{j2\pi pt} = \{F_p\}_p^P. \quad (3)$$

- “ $\|\cdot\|$ ” will refer to the quadratic norm,

$$\|\{v_p\}_p^P\| = \sqrt{\sum_{p=0}^{P-1} |v_p|^2}. \quad (4)$$

- δ_p will be Kronecker's delta: $\delta_0 = 1$ and $\delta_p = 0$ if $p \neq 0$.
- The arrow “ \rightarrow ” will denote a replacement in a given expression.

II. NFFT TYPES AND LAGRANGE INTERPOLATION FORMULA

The NFFT methods perform the same operations as the FFT or IFFT but allow for nonuniform sampling or evaluation grids. Basically, there exist five NFFT types, that depend on which grid is nonuniform. Following the classification in [8], the type-1 NFFT transforms a nonuniform into a uniform grid, and can be viewed as a method to sample regularly the spectrum of a nonuniform delta train like

$$\alpha(t) \equiv \sum_{p=1}^Q a_p \delta(t - t_p), \quad (5)$$

where $\{a_{p+1}\}_p^Q$ and $\{t_{p+1}\}_p^Q$ are a complex and real vector respectively, with $0 \leq t_p < 1$. More precisely, if $A(f)$ denotes the spectrum of $\alpha(t)$,

$$A(f) \equiv \sum_{p=1}^Q a_p e^{-j2\pi f t_p}, \quad (6)$$

the type-1 NFFT computes $\{A(p)\}_p^P$ from $\{t_{p+1}\}_p^Q$ and $\{a_{p+1}\}_p^Q$ with complexity $O(P \log P + Q)$. Typically, it involves one gridding operation and one FFT, both with oversampling factor two. Fundamentally, the gridding operation consists of replacing the deltas in (5) with band-limited discrete pulses.

One of the basic applications of the type-1 NFFT is to evaluate nonuniform convolutions of the form

$$\gamma(t) \equiv \sum_{p=1}^Q a_p \lambda(t - t_p) \quad (7)$$

in a regular grid $\{q/P\}_q^P$, where $\lambda(t)$ is a trigonometric polynomial

$$\lambda(t) \equiv \sum_{p=0}^{R-1} \Lambda_p e^{j2\pi p t}, \quad (8)$$

and R is an integer multiple of P , ($R \equiv \eta P$ with integer $\eta \geq 1$). To see the relation between $\gamma(t)$ and the type-1 NFFT, note that $\gamma(t)$ is the convolution of $\lambda(t)$ with the delta train in the type-1 NFFT in (5),

$$\gamma(t) = \lambda(t) * \sum_{p=1}^Q a_p \delta(t - t_p) = \lambda(t) * \alpha(t), \quad (9)$$

Thus, we have that $\gamma(t)$ is the polynomial

$$\gamma(t) \equiv \sum_{p=0}^{R-1} \Lambda_p A(p) e^{j2\pi p t}, \quad (10)$$

in which $\{A(p)\}_p^R$ can be obtained by means of one type-1 NFFT with $P \rightarrow R$. So, we have from (10) that $\{\gamma(q/P)\}_q^P$ is the output of one inverse DFT,

$$\{\gamma(q/P)\}_q^P = P \text{IDFT} \left\{ \sum_{r=0}^{\eta-1} \Lambda_{Pr+p} B_{Pr+p} \right\}_p^P. \quad (11)$$

The type-1 NFFT is a basic tool in spectral estimation from nonuniform samples [11] and, additionally, has various application fields like synthetic aperture radar (SAR) imaging

[1], graphics processing [2], and magnetic resonance imaging (MRI) [12].

The type-2 NFFT is complementary to the type-1 NFFT in the sense that it converts a uniform into a nonuniform grid. Specifically, given a trigonometric polynomial of the form

$$s(t) \equiv \sum_{p=0}^{P-1} S_p e^{j2\pi p t}, \quad (12)$$

where $\{S_p\}_p^P$ is a complex vector, the type-2 NFFT computes $s(t)$ at Q arbitrary instants $\{t_{p+1}\}_p^Q$ with complexity $O(P \log P + Q)$. Usually, it involves one weighting of the vector $\{S_p\}_p^P$, followed by one inverse FFT, and one final interpolation operation. This last operation consists of a short summation extended to the samples close to the interpolation instants t_{p+1} , and all the processing is performed with oversampling factor two. As (12) shows, its basic use is signal interpolation from spectral samples. It has applications in antenna design [13], computational electromagnetics [14], array processing [15], among other fields.

The type-3 NFFT is a combination of the previous two types and is a nonuniform to nonuniform transformation. In short, given the instants $\{t_{p+1}\}_p^Q$ and coefficients $\{a_{p+1}\}_p^Q$ in (5), the type-3 NFFT computes $\{A(f_{r+1})\}_r^R$ for a given set of frequencies $\{f_{r+1}\}_r^R$ with complexity $O(P \log P + Q + R)$. It is applied in heat flow computation and MRI [16].

Finally, types 4 and 5 are the inverses of types 1 and 2 respectively, assuming $P = Q$, and are usually computed iteratively by means of the conjugate gradient (CG) method. Specifically, from (6) the type-4 NFFT inverts the linear system

$$A(p) = \sum_{q=1}^P a_q e^{-j2\pi p t_q}, \quad p = 1, \dots, P, \quad (13)$$

where the unknowns are $\{a_{q+1}\}_q^P$. The existing CG method for this NFFT type locates the solution of (13) iteratively and in each iteration requires a discrete convolution with complexity $O(P \log P)$. The number of iterations required is usually large. From (12), the type-5 NFFT inverts the system

$$s(t_q) = \sum_{p=0}^{P-1} S_p e^{j2\pi p t_q}, \quad q = 0, 1, \dots, P-1, \quad (14)$$

and the existing CG method's performance is similar to that for type 4. One basic use of this NFFT type is to re-sample a signal to a regular grid [17]. Types 4 and 5 appear in computational electromagnetics [3], MRI [12], and spectral estimation [11], among other fields.

The purpose of this paper is to present two non-iterative methods for the type-4 and type-5 NFFTs in the one-dimensional case, which are significantly less expensive computationally than the state-of-the-art methods. They are based on the properties of the Lagrange formula

$$s(t) = \sum_{p=1}^P s(t_p) \frac{L(e^{j2\pi t})}{L'(e^{j2\pi t_p})(e^{j2\pi t} - e^{j2\pi t_p})}, \quad (15)$$

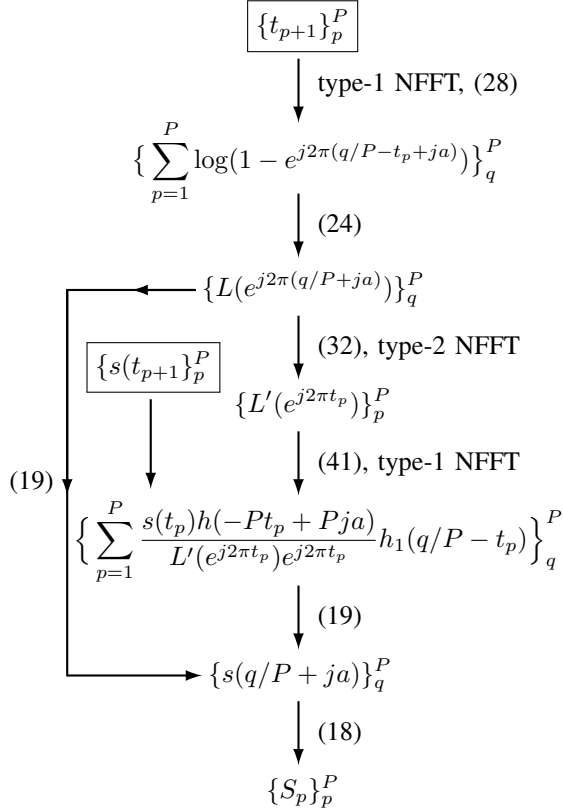


Fig. 1. Flow diagram of type-5 NFFT method.

where $L(e^{j2\pi t})$ is the kernel

$$L(e^{j2\pi t}) \equiv \prod_{p=1}^P e^{j2\pi t} - e^{j2\pi t_p}. \quad (16)$$

Fundamentally, they consist of writing (15) in terms of nonuniform convolutions and interpolation operations, which can be efficiently computed through type 1 and 2 NFFTs.

III. COMPUTATION OF THE TYPE-5 NFFT

In this section, we derive the proposed method for the type-5 NFFT, and in the next section the corresponding method for the type 4. We follow this order for simplicity, given that the latter is a simple corollary of the former. Since the derivations in this section are quite detailed, they have been split into separate sub-sections and in each of them one of the transitions in Fig. 1 is proved, starting with the last one and proceeding backward. Note that the inputs in that diagram (framed vectors) are the type-5 NFFT inputs, and the final output is the required vector of coefficients $\{S_p\}_p^P$.

Basically, the proposed type-5 NFFT method can be regarded as a procedure to evaluate the Lagrange formula in (15) simultaneously at all the instants in a regular grid $\{q/P + ja\}_q^P$, where ja is an imaginary shift, $a > 0$. This simultaneous evaluation is performed through convolutions like (7), type-2 NFFTs, DFTs, and other simpler operations. In the next sub-section, we start by relating the type-5 NFFT output $\{S_p\}_p^P$ with the sequence $\{s(q/P + ja)\}_q^P$, which is then computed in the sub-sections that follow.

A. Coefficients S_p from samples $s(q/P + ja)$

The final output $\{S_p\}_p^P$ can be computed from the sequence $\{s(q/P + ja)\}_q^P$ through one DFT, where ja is a fixed imaginary shift, $a > 0$. This is so because $s(t + ja)$ is just the output of passing $s(t)$ through an invertible filter, as the following equation reveals

$$s(t + ja) = \sum_{p=0}^{P-1} S_p e^{-2\pi p a} e^{j2\pi p t}. \quad (17)$$

The coefficients S_p are given by

$$\{S_p\}_p^P = \frac{1}{P} \{e^{2\pi p a}\}_p^P \odot \text{DFT}\{s(q/P + ja)\}_q^P, \quad (18)$$

and the problem comes down to computing $\{s(q/P + ja)\}_q^P$. As will be shown in the sequel, the shift ja is key in the method, given that it eliminates the singularities in two convolution kernels.

B. Sequence $s(q/P + ja)$ from Lagrange formula factors

In order to compute $\{s(q/P + ja)\}_q^P$, let us start by writing the Lagrange formula in (15) for $t = q/P + ja$ in the following way,

$$s(q/P + ja) = L(e^{j2\pi(q/P + ja)}) \cdot \sum_{p=1}^P \frac{s(t_p)}{L'(e^{j2\pi t_p})(e^{j2\pi(q/P + ja)} - e^{j2\pi t_p})}. \quad (19)$$

Note that this is the product of two terms and none of them has any poles or zeros for real t due to the shift ja . Thus, it is possible to compute $\{s(q/P + ja)\}_q^P$ by multiplying element-wise the sequences

$$\{L(e^{j2\pi(q/P + ja)})\}_q^P \quad (20)$$

and

$$\left\{ \sum_{p=1}^P \frac{s(t_p)}{L'(e^{j2\pi t_p})(e^{j2\pi(q/P + ja)} - e^{j2\pi t_p})} \right\}_q^P. \quad (21)$$

In the next sub-section, we show how to compute (20). Then, in Sec. III-E we address the computation of the samples $\{L'(e^{j2\pi t_p})\}_p^P$ appearing in (21), and in sub-section III-F the computation of the whole sequence (21).

C. Kernel samples $L(e^{j2\pi(q/P + ja)})$ from a nonuniform convolution.

Let us analyze the kernel $L(e^{j2\pi(t + ja)})$ in order to compute (20). From (16), we have that the logarithm of $L(e^{j2\pi(t + ja)})$ is the following:

$$\begin{aligned} \log L(e^{j2\pi(t + ja)}) &= \sum_{p=1}^P \log(e^{j2\pi(t + ja)} - e^{j2\pi t_p}) \\ &= jP\pi + j2\pi \sum_{p=1}^P t_p + \sum_{p=1}^P \log(1 - e^{j2\pi(t - t_p + ja)}). \end{aligned} \quad (22)$$

Let $v(t)$ denote the last summation

$$v(t) \equiv \sum_{p=1}^P \log(1 - e^{j2\pi(t-t_p+ja)}). \quad (23)$$

From the last two equations, we have that the kernel samples can be computed from the sequence $\{v(q/P)\}_q^P$ through the equation

$$L(e^{j2\pi(q/P+ja)}) = \exp\left(jP\pi + j2\pi \sum_{p=1}^P t_p + v(q/P)\right). \quad (24)$$

D. Computation of $v(q/P)$.

Let us analyze the definition of $v(t)$ in (23) in order to compute $\{v(q/P)\}_q^P$. The summation in that definition can be written as the convolution of a delta train with the kernel $\log(1 - e^{j2\pi(t+ja)})$,

$$\begin{aligned} v(t) &= \log(1 - e^{j2\pi(t+ja)}) * \sum_{p=1}^P \delta(t - t_p) \\ &= \log(1 - e^{j2\pi(t+ja)}) * z(t), \end{aligned} \quad (25)$$

where

$$z(t) \equiv \sum_{p=1}^P \delta(t - t_p). \quad (26)$$

Besides, the coefficients of $\log(1 - e^{j2\pi(t+ja)})$ decay with exponential trend as its Fourier series reveals,

$$\log(1 - e^{j2\pi(t+ja)}) = - \sum_{p=1}^{\infty} \frac{1}{p} e^{-2\pi pa} e^{j2\pi pt}. \quad (27)$$

Therefore, in (25) we may replace $\log(1 - e^{j2\pi(t+ja)})$ with this last series, but truncated at an index ensuring a negligible approximation error. Specifically, we have the approximation

$$v(t) \approx g(t) * z(t), \quad (28)$$

where

$$g(t) \equiv - \sum_{p=1}^{R-1} \frac{1}{p} e^{-2\pi pa} e^{j2\pi pt} \quad (29)$$

and R is the index of the first neglected coefficient. For simplicity, we take R equal to an integer multiple of P , $R = \eta P$, (integer $\eta \geq 1$). (28) is a nonuniform convolution of the form in (7). Thus, we may compute $\{v(q/P)\}_q^P$ using the procedure already described for (7) with $\lambda(t) \rightarrow g(t)$, $a_p \rightarrow 1$, and $R \rightarrow \eta P$.

E. Derivative samples $L'(e^{j2\pi t_p})$ from kernel samples $L(e^{j2\pi(q/P+ja)})$

Let $\{L_p\}_p^{P+1}$ denote the set of coefficients of $L(e^{j2\pi t})$ and note the following straight-forward equations

$$\text{Coef } L'(e^{j2\pi t}) = \{(p+1)L_{p+1}\}_p^P \quad (30)$$

$$\text{Coef } L(e^{j2\pi(t+ja)}) = \{e^{-2\pi pa} L_p\}_p^{P+1}. \quad (31)$$

The last equation implies that the DFT of $\{L(e^{j2\pi(q/P+ja)})\}_q^P$ gives the coefficients $\{L_p\}_p^P$ with a weighting. More precisely,

we have to take the DFT of $\{L(e^{j2\pi(q/P+ja)})\}_q^P$, remove the aliasing at $p = 0$, and compensate the factor $e^{-2\pi pa}$ in (31). The result is the following equation

$$\{L_p\}_p^P = \frac{1}{P} \left(\text{DFT}\{L(e^{j2\pi(q/P+ja)})\}_q^P - P e^{-2\pi Pa} \{\delta_p\}_p^P \right) \odot \{e^{2\pi pa}\}_p^P. \quad (32)$$

Once the coefficients $\{L_p\}_p^P$ are known, and noting that $L_P = 1$, we may obtain $\{L'(e^{j2\pi t_p})\}_p^P$ by applying a type-2 NFFT to $\{(p+1)L_{p+1}\}_p^P$, due to (30).

F. Summation in (21) assuming known samples $L'(e^{j2\pi t_p})$

In order to compute the summation in (21), let us first define the kernel

$$h(t) \equiv \frac{1}{e^{j2\pi t} - 1}, \quad (33)$$

and re-write the summation in that equation in the following way

$$\sum_{p=1}^P \frac{s(t_p)}{L'(e^{j2\pi t_p}) e^{j2\pi t_p}} h(q/P - t_p + ja) \quad (34)$$

$$= h(t + ja) * \sum_{p=1}^P \frac{s(t_p)}{L'(e^{j2\pi t_p}) e^{j2\pi t_p}} \delta(q/P - t_p). \quad (35)$$

Note that this expression resembles that in Sec. III-D, Eq. (25), for the computation of the samples $\{v(q/P)\}_q^P$. In it, we have the convolution of a signal $h(t + ja)$ with a nonuniform delta train. Besides, $h(t + ja)$ has infinite bandwidth, as can be readily seen in its Fourier series

$$h(t + ja) = - \sum_{r=0}^{\infty} e^{-2\pi ra} e^{j2\pi rt}, \quad (36)$$

and the delta train coefficients are known, given that $\{L'(e^{j2\pi t_p})\}_p^P$ has been pre-computed in Sec. III-E. However, (34) is a simpler case because $h(t + ja)$ can be replaced by a band-limited kernel, and this allows us to employ a type-1 NFFT without any truncation. To see this point, let us insert a term $h(q - Pt_p + Pja)$ in the summand in (34) and operate as follows:

$$\begin{aligned} & \frac{s(t_p)}{L'(e^{j2\pi t_p}) e^{j2\pi t_p}} h(q/P - t_p + ja) \\ &= \frac{s(t_p) h(q - Pt_p + Pja)}{L'(e^{j2\pi t_p}) e^{j2\pi t_p}} \cdot \frac{h(q/P - t_p + ja)}{h(q - Pt_p + Pja)}. \end{aligned} \quad (37)$$

Note that in the first fraction we may simplify

$$h(q - Pt_p + Pja) = h(-Pt_p + Pja), \quad (38)$$

and the second fraction is equal to $h_1(q/P - t_p)$, where $h_1(t)$ is a new band-limited pulse, defined by

$$\begin{aligned} h_1(t) &\equiv \frac{h(t + ja)}{h(P(t + ja))} = \frac{e^{j2\pi P(t+ja)} - 1}{e^{j2\pi(t+ja)} - 1} \\ &= \sum_{p=0}^{P-1} e^{-2\pi pa} e^{j2\pi pt}. \end{aligned} \quad (39)$$

of (27) requires a negligible ratio between the first and last summand of (29). Thus, if μ denotes this last ratio, defined by

$$\mu \equiv \frac{e^{-2\pi(\eta P-1)a}}{\eta P-1}, \quad (48)$$

then a and η should be selected to ensure μ is close to the working precision. So, we may see from (48) that, in rough terms, the product $aP\eta$ must be sufficiently large, and this can be achieved by either increasing a or η . However, an increase in a produces a strong attenuation in the computation of the coefficients $\{L_p\}_p^P$ in (32) due to the vector $\{e^{2\pi p a}\}_p^P$. And an increase in η seems suitable for truncating (27) while reducing the damping effect in (32), but there is a corresponding increase in the computational burden, because the computation of $\{v(q/P)\}_q^P$ involves a type-1 NFFT of size ηP .

A simple way to overcome this situation consists of selecting a and μ that produce an inaccurate result, and then applying the method twice, first to the input sequences $\{t_{p+1}\}_p^P$, $\{a_{p+1}\}_p^P$, and then to $\{t_{p+1}\}_p^P$ and the residual error, which can be computed through a type-1 or type-2 NFFT. More precisely, suppose we require to compute the type-4 NFFT, but the method available produces an error $\{\Phi_{0,p}\}_p^P$,

$$\{t_{p+1}\}_p^P, \{a_{p+1}\}_p^P \xrightarrow{\text{type-4 method}} \{A(p) + \Phi_{0,p}\}_p^P. \quad (49)$$

Also, suppose that the accuracy of this method is $\epsilon < 1$ in the sense that

$$\|\{\Phi_{0,p}\}_p^P\| < \epsilon \|\{A(p)\}_p^P\| \quad (50)$$

for any possible input sequence $\{a_{p+1}\}_p^P$. Then, we may obtain a more accurate approximation of $\{A(p)\}_p^P$ in the following steps:

- 1) Compute the type-1 NFFT of the right-hand side of (49),

$$\{t_{p+1}\}_p^P, \{A(p) + \Phi_{0,p}\}_p^P \xrightarrow{\text{type-1 NFFT}} \{a_{p+1} + \phi_{0,p+1}\}_p^P,$$

where $\{\phi_{0,p+1}\}_p^P$ is the type-1 NFFT of $\{\Phi_{0,p}\}_p^P$.

- 2) Subtract $\{a_{p+1}\}_p^P$ to the last output to obtain $\{\phi_{0,p+1}\}_p^P$.
- 3) Apply the type-4 method to $\{\phi_{0,p+1}\}_p^P$,

$$\{\phi_{0,p+1}\}_p^P \xrightarrow{\text{type-4 method}} \{\Phi_{0,p} + \Phi_{1,p}\}_p^P$$

where $\{\Phi_{1,p}\}_p^P$ is a new residual error. Now we have

$$\|\{\Phi_{1,p}\}_p^P\| < \epsilon^2 \|\{A(p)\}_p^P\|. \quad (51)$$

- 4) Subtract the last sequence from the output of (50). The result is $\{A(p) - \Phi_{1,p}\}_p^P$ and we have doubled the accuracy due to (51).

A similar refinement can be applied to the type-5 NFFT.

VI. NUMERICAL EXAMPLES

In this section, we evaluate the proposed NFFT methods in terms of round-off error and computational burden. We only present results for the type-4 NFFT, though they also represent the corresponding performance of the type-5 NFFT. Actually, the figures that follow and those for the type-5 NFFT are identical and, therefore, it is redundant to repeat them in

Operation	Flops
Real sum	1
Complex sum	2
Real multiplication	1
Complex multiplication	6
Complex exponential	7
Size- N FFT, IFFT	$5N \log_2 N$

Fig. 3. Flop counts for basic operations.

the present paper. This identical performance is due to the relation between the linear systems in (13) and (14). As can be readily inferred from these last equations, the linear system solved by the types 4 and 5 NFFT form a dual pair, i.e., if we take the Hermitian of the type-4 linear system matrix we obtain the corresponding type-5 matrix. A consequence of this duality is that methods like Gaussian elimination and conjugate gradient have identical round-off error performance and computational burden for both types. And this is also true for the NFFT methods proposed in this paper. As can be easily deduced from Figs. 1 and 2, the type-4 and type-5 evaluation procedures are the same except for a small modification that involves no change in either the round-off error performance or computational burden.

We have evaluated the performance of the type-4 (and type-5) NFFT method in the following setup:

- *Monte Carlo trials.* The figures have been generated from just 10 Monte Carlo trial, given that the large values of P produce low-variance round-off error estimates. In these trials, the sampling instants $\{t_{p+1}\}_p^P$ were obtained by shifting the elements of a regular grid with spacing $1/P$. These shifts were independent and had uniform distribution in the interval $[0, 0.6/P]$. The amplitudes $\{a_{p+1}\}_p^P$ were independent complex Gaussian samples of zero mean and variance one.
- *Numerical methods.* We have used the following methods:
 - GE: Computation based on inverting the associated linear system through Gaussian elimination, [9, Ch. 3].
 - CG: The same inversion but using the conjugate gradient method [18, p. 73].
 - NFFT: Method proposed in this paper in Sec. IV.
 - R-NFFT: Previous method with refinement (Sec. V).
- *Computational burden.* We have measured the computational burden in floating-point operations (flops), following the counts in Fig. 3 for basic operations.
- *Round-off error measure.* Given a true vector $\{A(p)\}_p^P$ and an interpolated vector $\{\tilde{A}_p\}_p^P$, the error measure has been

$$\frac{\|\{A(p) - \tilde{A}_p\}_p^P\|}{\|\{A(p)\}_p^P\|}.$$

Fig. 4 shows the round-off error of the NFFT method versus the attenuation μ in (48) for several oversampling factors η and $P = 1024$. This figure also includes the round-off error of the GE and CG methods as benchmarks. Note that with $\eta = 1$, the achievable round-off error is around -130 dB, which is a value significantly larger than the GE, CG benchmarks. However,

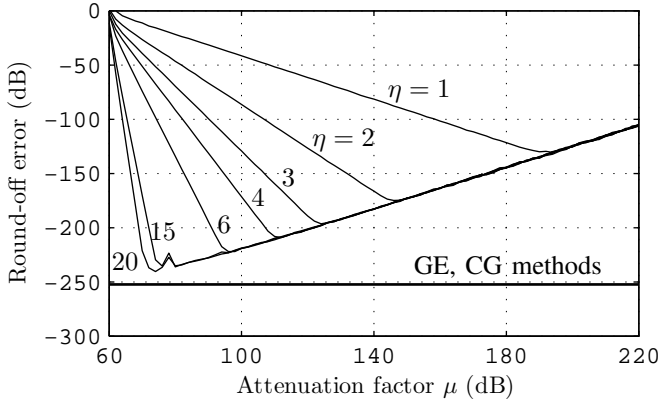


Fig. 4. Round-off error versus attenuation factor (μ) for several oversampling factors (η) for the type-4 NFFT method.

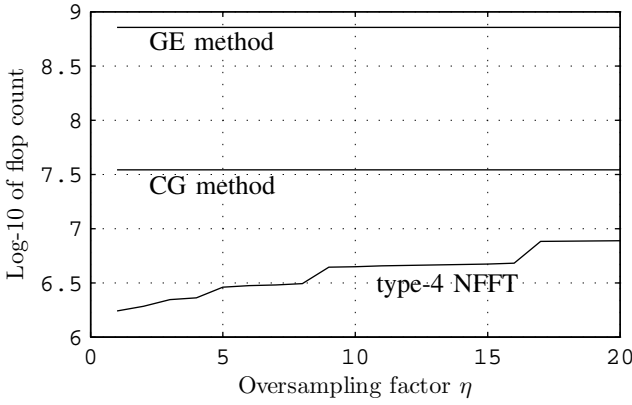


Fig. 5. Computational burden of type-4 NFFT method, measured in floating point operations (flops), versus oversampling factor η . 5

this error decreases with η . With $\eta = 6$ it is around -220 dB and can be reduced by increasing η to a value only 10 dB above the previous benchmarks. As Fig. 5 shows, this decrease is obtained at the expense of a higher computational burden but, by far, the type-4 NFFT is the cheapest computationally. Actually, its computational burden is more than factor 10 smaller that the CG method's complexity for $\eta = 6$.

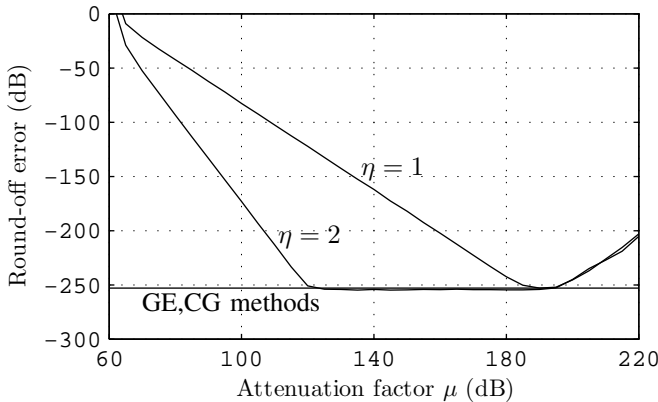


Fig. 6. Round-off error versus attenuation factor (μ) for two oversampling factors (η) for the iterated type-4 NFFT method.

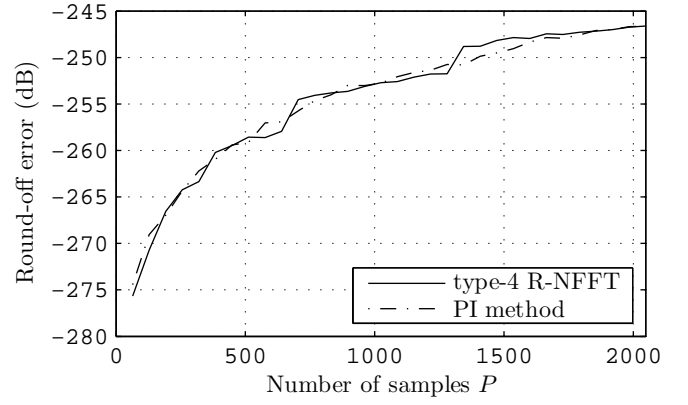


Fig. 7. Round-off error versus number of samples P for the type-4 R-NFFT method.

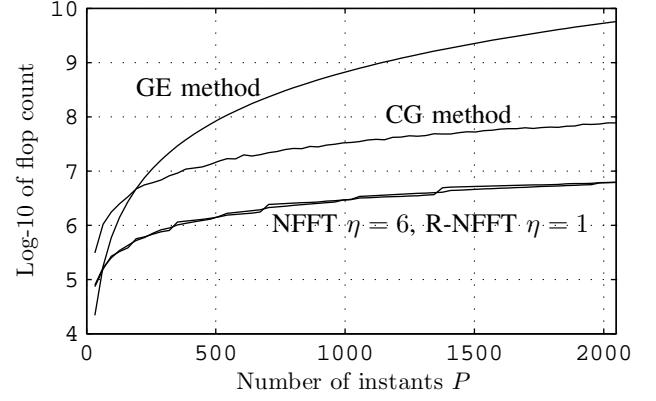


Fig. 8. Flop counts of four methods: GE, CG, type-4 NFFT with $\eta = 6$, and type-4 R-NFFT with $\eta = 1$.

Fig 6 shows the round-off error of the type-4 R-NFFT method versus the attenuation factor μ for $\eta = 1, 2$. Note that without any oversampling ($\eta = 1$), we may select an attenuation μ for which its round-off error is similar to that of the GE or CG methods. Fig. (7) shows the same round-off error but versus P for $\eta = 1$. For each abscissa in this figure, μ has been selected to minimize the round-off error. We can see that the type-4 R-NFFT method reaches the GE, CG benchmark for typical P values.

Finally, Fig. 8 shows the flop count of four methods: CG, GE, type-4 NFFT with $\eta = 6$, and type-4 R-NFFT with $\eta = 1$. Note that the last two methods roughly have the same flop count but, as shown in Fig. 4, the NFFT method is slightly above the GE, CG benchmark, while the R-NFFT method reaches it (Fig. 7).

VII. CONCLUSIONS

We have presented non-iterative methods for the type 4 and 5 NFFTs in the one-dimensional case, which are significantly less expensive computationally than the state-of-the-art methods like the Conjugate Gradient. The methods are based on expressing the Lagrange formula in terms of nonuniform convolutions that can be efficiently evaluated using the type 1 and 2 NFFTs. The paper contains several numerical examples

in which the proposed methods are compared with the Gaussian elimination (GE) and conjugate gradient (CG) methods in terms of round-off error and computational burden.

REFERENCES

- [1] F. Andersson, R. Moses, and F. Natterer, "Fast Fourier methods for synthetic aperture radar imaging," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 1, pp. 215–229, Jan 2012.
- [2] Susanne Kunis and Stefan Kunis, "The nonequispaced FFT on graphics processing units," *PAMM*, vol. 12, no. 1, pp. 7–10, 2012.
- [3] L. Zhou, Z.H. Fan, L. Mo, R.S. Chen, and D.X. Wang, "A fast inverse NUFFT algorithm for computational electromagnetics," in *IEEE Antennas and Propagation Society International Symposium*, July 2005, vol. 4B, pp. 160–163.
- [4] Jens Keiner, Stefan Kunis, and Daniel Potts, "Using NFFT 3—a software library for various nonequispaced fast Fourier transforms," *ACM Transactions on Mathematical Software*, vol. 36, no. 4, pp. 1–30, aug 2009.
- [5] D. Potts, *Schnelle Fourier-Transformationen für nichtäquidistante Daten und Anwendungen*, Ph.D. thesis, Lübeck University, 2003.
- [6] John J Benedetto, *Wavelets: mathematics and applications*, vol. 13, chapter 12, CRC press, 1993.
- [7] A. J. W. Duijndam and M. A. Schonewille, "Nonuniform Fast Fourier Transform," *Geophysics*, vol. 64, no. 2, pp. 539–551, 1999.
- [8] Alok Dutt and Vladimir Rokhlin, "Fast fourier transforms for nonequispaced data," *SIAM Journal on Scientific computing*, vol. 14, no. 6, pp. 1368–1393, 1993.
- [9] Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, fourth edition, 2013.
- [10] Mariano Gasca and Thomas Sauer, "Polynomial interpolation in several variables," *Advances in Computational Mathematics*, vol. 12, no. 4, pp. 377–410, 2001.
- [11] Prabhu Babu and Petre Stoica, "Spectral analysis of nonuniformly sampled data—a review," *Digital Signal Processing*, vol. 20, no. 2, pp. 359–378, 2010.
- [12] Jeffrey A. Fessler and Bradley P. Sutton, "Nonuniform fast Fourier transform using mini-max interpolation," *IEEE Transactions on Signal Processing*, vol. 51, pp. 560–574, Feb. 2003.
- [13] Amedeo Capozzoli, Claudio Curcio, Angelo Liseno, and Annalisa Riccardi, "Selecting parameters of type-3 NUFFTs to control accuracy in MoM methods," in *Antennas and Propagation (EuCAP), 2014 8th European Conference on*, IEEE, 2014, pp. 1157–1161.
- [14] Q.H. Liu and N. Nguyen, "An accurate algorithm for nonuniform fast fourier transforms (NUFFT's)," *IEEE Microwave and Guided Wave Letters*, vol. 8, no. 1, pp. 18–20, Jan 1998.
- [15] J. Selva, "An efficient Newton-type method for the computation of ML estimators in a Uniform Linear Array," *IEEE Transactions on Signal Processing*, vol. 53, no. 6, pp. 2036–2045, June 2005.
- [16] June-Yub Lee and Leslie Greengard, "The type 3 nonuniform FFT and its applications," *Journal of Computational Physics*, vol. 206, no. 1, pp. 1–5, 2005.
- [17] J. Selva, "FFT interpolation from nonuniform samples lying in a regular grid," *Signal Processing, IEEE Transactions on*, vol. 63, no. 11, pp. 2826–2834, June 2015.
- [18] S. Kunis, *Nonequispaced FFT, generalisation and inversion*, Ph.D. thesis, Technisch-Naturwissenschaftlichen Fakultät, Universität zu Lübeck, 2006.